

TUGAS KOMPUTASI PARALEL

Dosen Pengampu : Asril Adi Sunarto, M.Kom.



Muhammad Azril Alfarizi (2230511044)

Teknik Informatika 7A

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH SUKABUMI
TAHUN 2024/2025

Kode Program :

```
#include <mpi.h>
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char** argv) {
    int rank, size;
    long long i;
    // jumlah elemen. Sesuaikan jika ingin hasil seperti contoh: N = 1_000_000
-> sum = 500000500000
    long long N = 1000000LL;
    if (argc > 1) {
        N = atol(argv[1]);
        if (N <= 0) N = 1000000LL;
    }

    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);

    long long chunk = N / size;
    long long start_i = rank * chunk + 1; // 1-based
    long long end_i = (rank == size - 1) ? N : start_i + chunk - 1;

    double t_start = MPI_Wtime();

    // Hitung sum lokal
    long long local_sum = 0;
    for (i = start_i; i <= end_i; ++i) {
        local_sum += i;
    }

    // Lakukan binary-tree reduction dengan point-to-point (MPI_Send/MPI_Recv)
    long long recv_sum;
    int partner;
    int step;
    for (step = 1; step < size; step <= 1) {
        if ((rank % (2*step)) == 0) {
            partner = rank + step;
            if (partner < size) {
                MPI_Recv(&recv_sum, 1, MPI_LONG_LONG_INT, partner, 0,
MPI_COMM_WORLD, MPI_STATUS_IGNORE);
                local_sum += recv_sum;
            }
        } else {
            partner = rank - step;
            MPI_Send(&local_sum, 1, MPI_LONG_LONG_INT, partner, 0,
MPI_COMM_WORLD);
```

```

        break; // setelah mengirim keluar dari loop
    }

double t_end = MPI_Wtime();

if (rank == 0) {
    printf("\n--- Hasil Eksekusi ---\n");
    printf("Jumlah elemen N      : %lld\n", N);
    printf("Hasil Akhir (Total Sum): %lld\n", local_sum);
    printf("Waktu Eksekusi       : %f detik\n", t_end - t_start);
    printf("Kompleksitas Total   : O(N/P + log P) (komputasi: O(N/P),\nreduksi tree: O(log P))\n");
}

MPI_Finalize();
return 0;
}

```

Analisa Hasil :

- **Data**

Jumlah Prosesor (-np)	Waktu Eksekusi (detik)	Hasil Akhir
1	0.003063	500000500000
2	0.003042	500000500000
3	0.001198	500000500000
4	0.001067	500000500000
5	0.000825	500000500000

- **Analisis performa**

-np	Waktu (s)	Speedup (W_1/W_p)	Efisiensi (%)
1	0.003063	1.00	100
2	0.003042	1.01	50
3	0.001198	2.56	85
4	0.001067	2.87	72
5	0.000825	3.71	74

Rumus :

- $Speedup = \frac{W(1)}{W(p)}$
- $Efisiensi = \frac{speedup}{p} \times 100\%$

HASIL :

```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS
● jii@DESKTOP-0F2UHGS:/mnt/c/Users/Ajil/Documents/Komputasi-Paralel/Tugas$ mpirun -np 1 tugas
--- Hasil Eksekusi ---
Jumlah elemen N      : 1000000
Hasil Akhir (Total Sum): 500000500000
Waktu Eksekusi       : 0.003063 detik
Kompleksitas Total   : O(NP + log P) (komputasi: O(N/P), reduksi tree: O(log P))
● jii@DESKTOP-0F2UHGS:/mnt/c/Users/Ajil/Documents/Komputasi-Paralel/Tugas$ mpirun -np 2 tugas
--- Hasil Eksekusi ---
Jumlah elemen N      : 1000000
Hasil Akhir (Total Sum): 500000500000
Waktu Eksekusi       : 0.003042 detik
Kompleksitas Total   : O(NP + log P) (komputasi: O(N/P), reduksi tree: O(log P))
● jii@DESKTOP-0F2UHGS:/mnt/c/Users/Ajil/Documents/Komputasi-Paralel/Tugas$ mpirun -oversubscribe -np 3 tugas
--- Hasil Eksekusi ---
Jumlah elemen N      : 1000000
Hasil Akhir (Total Sum): 500000500000
Waktu Eksekusi       : 0.001198 detik
Kompleksitas Total   : O(NP + log P) (komputasi: O(N/P), reduksi tree: O(log P))
● jii@DESKTOP-0F2UHGS:/mnt/c/Users/Ajil/Documents/Komputasi-Paralel/Tugas$ mpirun --oversubscribe -np 4 tugas
--- Hasil Eksekusi ---
Jumlah elemen N      : 1000000
Hasil Akhir (Total Sum): 500000500000
Waktu Eksekusi       : 0.001067 detik
Kompleksitas Total   : O(NP + log P) (komputasi: O(N/P), reduksi tree: O(log P))
● jii@DESKTOP-0F2UHGS:/mnt/c/Users/Ajil/Documents/Komputasi-Paralel/Tugas$ mpirun --oversubscribe -np 5 tugas
--- Hasil Eksekusi ---
Jumlah elemen N      : 1000000
Hasil Akhir (Total Sum): 500000500000
Waktu Eksekusi       : 0.000825 detik
Kompleksitas Total   : O(NP + log P) (komputasi: O(N/P), reduksi tree: O(log P))
● jii@DESKTOP-0F2UHGS:/mnt/c/Users/Ajil/Documents/Komputasi-Paralel/Tugas$
```